Towards cyber-physical systems: Distributed model predictive control

LI ShaoYuan* , ZHENG Yi & WEI YongSong

Key Laboratory of System Control and Information Processing, Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China

Received April 19, 2015; accepted June 22, 2015

Abstract In this paper, we provide a review of recent results in the design of distributed model predictive control (DMPC). DMPC not only inherits the advantage of model predictive control but also has characteristics of distributed control framework. We review the work on DMPC from two aspects: unconstrained DMPC and the design methods of stabilized DMPC with constraints. Finally, some proposed algorithms are illustrated through two industrial processes.

Keywords Cyber-physical system; Model predictive control; Distributed model predictive control; Plant-wide system; Iterative; Networked

doi: 10.16262/j.cnki.1005-0841.2015.03.001

1 Introduction

The cyber-physical systems (CPSs) refer to a new generation system with integrated computational and physical capabilities that can interact with humans through many modalities [1]. One of the three characteristics of CPSs [2] is that the CPSs are a system of systems which consist of many subsystems that can stand alone in an individual manner. Among those subsystems, there are novel interactions between control, communication, and computation. In this way, the structure of the global system is a classical distributed one.

For this class of large scale systems with hundreds or thousands of input and output variables (e.g., power and energy network, transportation system, and large chemical processes, etc.), it is often impractical to apply the classical centralized MPC, where a control agent is able to acquire the information of the global system and could obtain a good global performance, to large scale systems for three reasons: first, there are hundreds of inputs and outputs which require tremendous computational efforts in online implementation; second, when the centralized controller fails, the entire system is out of control and the control integrity cannot be guaranteed if a control component does not work; third, in some cases, e.g. in multi-intelligent vehicle system, the global information is unavailable to any controller. Thus, the disturbed model predictive control appeared and has gradually substituted the centralized MPC.

The distributed Predictive not only inherits the advantages of model predictive control in directly handling constraints and good optimization performance, but also has the characteristics of distributed control framework with less computational burden, high flexible, good error tolerance and no global information requirements [3, 4]. Using distribute predictive control, the future state information of each subsystem is

* Corresponding author. E-mail: syli@sjtu.edu.cn.

able to feed into its interacted subsystem-based MPC before satisfying the versatile control objective, e. g. large lag system, and more restricting control performance requirements. The advantages of distributed predictive control are as follows.

(1) Its underlying ideas are easy to understand: The distributed predictive control is the distributed implementation of a set of predictive controllers, and these predictive controllers consider the feed forward information from the predictive controllers which correspond to the subsystems they interacted with.

(2) The local predictive control can deal with equipment and safety constraints in routine.

(3) The local predictive control handles multivariable control problems naturally. It is more powerful than PID control, even for single loops without constraints. It is not much more difficult to tune, even on difficult loops such as those containing long-time delay.

(4) It allows operation closer to constraints compared with conventional control, which frequently leads to more profitable operation.

(5) Since the centralized predictive control is decomposed into many small-scaled predictive controllers, the computational efforts in each small-scaled predictive control are less than those used for solving the centralized predictive control.

(6) If one or several errors occurred in a subsystem, the other subsystem-based predictive controllers are able to go on with their work without any disturbance. It has good error tolerance.

(7) If new subsystems are added to the current system, it is not necessary to modify every local predictive control. What should be done is just alter the predictive control whose corresponding subsystem is interacting with the newly-added subsystems. The distributed predictive control owes high flexibility to the system structure.

(8) The plug-in and plug-out are also able to be realized if a suitable algorithm and an appropriate program are designed.

Due to these advantages, the distributed predictive control gradually takes the place of centralized predictive control for plant-wide systems. However, the optimization performance of distributed predictive control, in most cases, is not as good as that of centralized predictive control [3—9]. Thus, many different coordinating strategies are proposed to solve this problem [3—18]. In most cases, the coordinating strategies are very important for the performance of the closed-loop systems.

To improve the global performance of the DMPC, several coordination strategies have appeared in the literature, and can be classified according to the information exchange protocol needed and the type of cost function which is optimized [19]. There are two classes of distributed predictive controls if we catalog the distributed predictive control by information exchange protocol-non-iterative algorithm and iterative algorithm.

(1) Non-iterative-based algorithm: in this kind of distributed predictive control, each local predictive control only communicates once with other local predictive control within every single control period, and solves the local control law once in a control period [11, 18, 20, 21].

(2) Iterative-based algorithm: this kind of distributed predictive control assumes that the network communication resources are abundant for supporting each local predictive control communicating with other interacted local predictive control many times in a single control period. And the time taken in communicating is so little that it could be ignored comparing with the control period. Each local predictive control solves its optimal control law based on the presumed control sequence before transforming this control law to its interacted local predictive controllers. After that, each local predictive controller solves the new optimal control law based on the optimal control laws of its neighbors tackled last time. This process will then

be repeated until the iteration stopping conditions are satisfied [5, 10, 22—24].

The non-iterative algorithms consume less communication resources than iterative algorithm, and have a faster computation speed in comparison with the iterative algorithm. The iterative algorithms are able to achieve better global performance than the non-iterative algorithms.

If we catalog the distributed predictive control by the type of cost function each local predictive control optimized, there are three types of distributed predictive control methods: Local Cost Optimization MPC [25], Cooperative Distributed MPC [22] and Networked Distributed MPC [26].

(1) Local Cost Optimization Based MPC (LCO-DMPC): Distributed algorithms where each subsystem-based controller minimizes the cost function of its own subsystem [25].

$$J_i(k) = \| \mathbf{x}_i(k+N) \|^2_{\mathbf{P}_i} + \sum_{s=0}^{N-1} (\| \mathbf{x}_i(k+s) \|^2_{\mathbf{Q}_i} + \| \mathbf{u}_i(k+s) \|^2_{\mathbf{R}_i}).$$

When computing the optimal solution, each local controller exchanges state estimation with the neighboring subsystems to improve the performance of the local subsystem. This method is simple and very convenient for implementation. An extension of this stabilizing DMPC with input constraint for nonlinear continuous systems [27, 28], and a stabilizing DMPC with inputs and states constraints [29] also have been reported.

Xi [30] developed an iterative algorithm for distributed MPC based on Nash optimality. The whole system will arrive at Nash equilibrium if the convergence condition of the algorithm is satisfied.

(2) Cooperative Distributed MPC (C-DMPC): To improve the global performance, distributed algorithms, where each local controller minimizes a global cost function were proposed [8, 10, 18, 22, 31].

$$J_i(k) = \sum_{j \in \mathcal{P}} J_j(k).$$

In this method, each subsystem-based MPC exchanges information with all other subsystems, and some iterative stabilizing designs proposed which take advantage of the model of whole system are used in each subsystem-based MPC. This strategy may result in a better performance but consumes much more communication resources, in comparison with the method described in LCO-DMPC.

(3) Networked Distributed MPC with information constraints (N-DMPC): To balance the performance, communication cost, and the complexity of the DMPC algorithm, the strategy that each subsystem-based controller only minimizes its own cost function and those of the subsystems that its own subsystem directly impacts on was recently proposed [20, 23, 26].

$$\bar{J}_i(k) = \sum_{j \in \mathcal{P}_i} J_j(k),$$

where $\mathcal{P}_i = \{ j : j \in \mathcal{P}_{-i} \ \ or\ j = i \}$ is the set of subscripts of the downstream subsystems of subsystem, and that is the region impacted on by subsystems. The resulting control algorithm is termed as an impacted-region cost optimization based on DMPC (ICO-DMPC) [32—34] or network distributed MPC (N-DMPC) with communication constraints. It could achieve a better performance than the first method, and its communication burden is far less than the second method. Clearly, this coordination strategy same as reported before [20, 23, 26] is a preferable method to trade-off the communication burden and the global performance.

The methods described in [27, 35] are proposed for a set of decoupled subsystems, and the extension of [27] could handle systems with weakly interacting subsystem dynamics [28]. There is no absolute priority among these different distributed predictive controllers. We could select different algorithms according to their purpose on employing control system.

2 Preliminaries

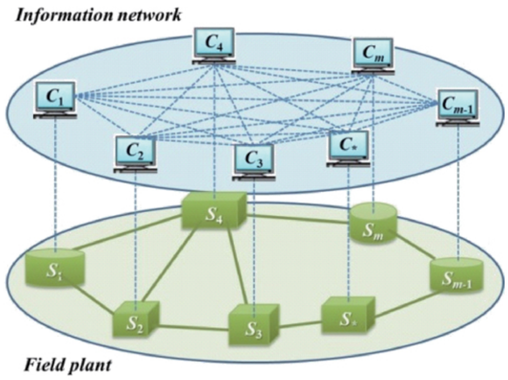


Figure 1 The schematic of distributed systems.

The distributed system (Figure 1) is composed of many interacting subsystems, each of which is controlled by a subsystem-based controller, which in turn is able to exchange information with other subsystem-based controllers. Suppose that the distributed system $\mathcal{S}$ is composed of m discrete-time linear subsystems $\mathcal{S}_i$, $i \in \mathcal{P} = \{1, 2, \cdots, m\}$ and m controllers $\mathcal{C}_i$, $i \in \mathcal{P} = \{1, 2, \cdots, m\}$. Let the subsystems interact with each other through their states. If subsystem $\mathcal{S}_i$ is affected by $\mathcal{S}_j$ $i \in \mathcal{P}$ $j \in \mathcal{P}$, subsystem $\mathcal{S}_i$ is a downstream subsystem of subsystem $\mathcal{S}_j$ and subsystem $\mathcal{S}_j$ is an upstream system of $\mathcal{S}_j$. Let $\mathcal{P}_{+i}$ denote the set of the subscripts of the upstream systems of $\mathcal{S}_i$, and $\mathcal{P}_{-i}$ the set of the subscripts of the downstream systems of $\mathcal{S}_i$. Then, subsystem $\mathcal{S}_i$ can be expressed as:

$$\begin{cases} \mathbf{x}_i(k+1) = \mathbf{A}_{ii}\,\mathbf{x}_i(k) + \mathbf{B}_{ii}\,\mathbf{u}_i(k) + \sum_{j \in \mathcal{P}_i^u} (\mathbf{A}_{ij}\,\mathbf{x}_j(k) + \mathbf{B}_{ij}\,\mathbf{u}_j(k)), \\ \mathbf{y}_i(k+1) = \mathbf{C}_{ii}\,\mathbf{x}_i(k) + \mathbf{C}_{ij}\,\mathbf{x}_j(k), \end{cases} \tag{1}$$

where $\mathbf{x}_i(k) \in \mathbb{R}^{n_{xi}}$, $\mathbf{u}_i(k) \in \mathcal{U}_i \subset \mathbb{R}^{n_{ui}}$, and $\mathbf{y}_i(k) \in \mathbb{R}^{n_{yi}}$ are respectively the local state, input and output vectors, and $\mathcal{U}_i$ is the feasible set of the input $\mathbf{u}_i(k)$, which is used to bound the input according to the physical constraints on the actuators, the control requirements or the characteristics of the plant A. A non-zero matrix $\mathbf{A}_{ij}$, that is, $j \in \mathcal{P}_{+i}$, indicates that $\mathcal{S}_i$ is affected by $\mathcal{S}_j$. In the concatenated vector form, the system dynamics can be written as:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \\ \mathbf{y}(k+1) = \mathbf{C}\mathbf{x}(k), \end{cases} \tag{2}$$

where

$$\begin{aligned} \mathbf{x}(k) &= [\mathbf{x}_1^{\mathrm{T}}(k) \quad \mathbf{x}_2^{\mathrm{T}}(k) \quad \cdots \quad \mathbf{x}_m^{\mathrm{T}}(k)]^{\mathrm{T}} \in \mathbb{R}^{n_x}, \\ \mathbf{u}(k) &= [\mathbf{u}_1^{\mathrm{T}}(k) \quad \mathbf{u}_2^{\mathrm{T}}(k) \quad \cdots \quad \mathbf{u}_m^{\mathrm{T}}(k)]^{\mathrm{T}} \in \mathcal{U}_i \subset \mathbb{R}^{n_u}, \\ \mathbf{y}(k) &= [\mathbf{y}_1^{\mathrm{T}}(k) \quad \mathbf{y}_2^{\mathrm{T}}(k) \quad \cdots \quad \mathbf{y}_m^{\mathrm{T}}(k)]^{\mathrm{T}} \in \mathbb{R}^{n_y}, \end{aligned}$$

are the concatenated state, control input and output vectors of the overall system S, respectively. Also, $\mathbf{u}(k) \in \mathcal{U}_i = \mathcal{U}_{i1} \times \mathcal{U}_{i2} \times \cdots \times \mathcal{U}_{im}$. A, B and C are constant matrices of appropriate dimensions and are defined as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \cdots & \mathbf{A}_{1m} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \cdots & \mathbf{A}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{m1} & \mathbf{A}_{m2} & \cdots & \mathbf{A}_{mm} \end{bmatrix}^{\mathrm{T}},$$

$$\mathbf{B}=\begin{bmatrix}\mathbf{B}_{11} & \mathbf{B}_{12} & \cdots & \mathbf{B}_{1m}\\ \mathbf{B}_{21} & \mathbf{B}_{22} & \cdots & \mathbf{B}_{2m}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{B}_{m1} & \mathbf{B}_{m2} & \cdots & \mathbf{B}_{mm}\end{bmatrix}^{\mathrm{T}},$$

$$\mathbf{C}=\begin{bmatrix}\mathbf{C}_{11} & \mathbf{C}_{12} & \cdots & \mathbf{C}_{1m}\\ \mathbf{C}_{21} & \mathbf{C}_{22} & \cdots & \mathbf{C}_{2m}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{C}_{m1} & \mathbf{C}_{m2} & \cdots & \mathbf{C}_{mm}\end{bmatrix}^{\mathrm{T}}.$$

If there is only the state interacting term, we call the model state interacted model and it can be expressed as:

$$\mathbf{x}_i(k+1)=\mathbf{A}_{ii}\,\mathbf{x}_i(k)+\mathbf{B}_{ii}\,\mathbf{u}_i(k)+\sum_{j\in\mathcal{P}_{+i}}\mathbf{A}_{ij}\,\mathbf{x}_j(k). \tag{3}$$

Similarly, if there is only the input interacting term of input, we call the model input interacted model and express it as:

$$\mathbf{x}_i(k+1)=\mathbf{A}_{ii}\,\mathbf{x}_i(k)+\mathbf{B}_{ii}\,\mathbf{u}_i(k)+\sum_{j\in\mathcal{P}_i^u}\mathbf{B}_{ij}\,\mathbf{u}_j(k). \tag{4}$$

In fact, the state interacted model can be transformed into the input interacted model if a suitable transformation is employed. And shared states (states which belong to both interacted subsystems) are defined.

3 Unconstrained distributed MPC

3.1 The local cost optimization based distributed MPC

The control objective of this system is minimizing a global performance index at time k under the distributed control framework, and

$$J(k)=\sum_{i=1}^{m}\Big[\sum_{l=1}^{P}\|\,\mathbf{y}_i(k+l)-y_i^d(k+l)\,\|_{\mathbf{Q}_i}^2+\sum_{l=1}^{M}\|\,\Delta\mathbf{u}_i(k+l-1)\,\|_{\mathbf{R}_i}^2\Big], \tag{5}$$

where $\mathbf{Q}_i$ and $\mathbf{R}_i$ are weight matrices, $P, M\in\mathbb{N}$ are predictive, horizon, and control horizon respectively, and $P\geqslant M$, $\mathbf{y}_i^d$ is the set-point of subsystem $\mathcal{S}_i$, while $\Delta\mathbf{u}_i(k)=\mathbf{u}_i(k)-\Delta\mathbf{u}_i(k-1)$ is the input increment vector of subsystem $\mathcal{S}_i$. For the large scale system considered here, the global performance index (5) can be decomposed in terms of the local performance index for each subsystem S_i, $i=1, 2, \cdots, m$ [36].

$$J_i(k)=\sum_{l=1}^{P}\|\,\hat{\mathbf{y}}_i(k+l|k)-\mathbf{y}_i^d(k+l|k)\,\|_{\mathbf{Q}_i}^2+\sum_{l=1}^{M}\|\,\Delta\mathbf{u}_i(k+l-1|k)\,\|_{\mathbf{R}_i}^2. \tag{6}$$

The local control decision of S_i is computed by solving the optimization problem min $J_i(k)$ with local input/output variables and constraints in the distributed MPC based on the state (or input) estimations of neighbors at time $k-1$.

To predict the future state of current subsystem S_i, the only information that each controller C_i, $i=1, 2, \cdots, n$, needs is the future behavior of the subsystems S_i controlled by agents $\mathcal{C}_j\in\mathcal{P}_{+i}$. Similarly, C_i should broadcast the future behavior of the local variables only to the agents $\mathcal{C}_j\in\mathcal{P}_{-i}$. Then the states and outputs of the downstream neighbors in l-step ahead can be predicted by

$$\begin{cases}\hat{\mathbf{x}}_i(k+l\mid k)=\mathbf{A}_{ii}^l\,\hat{\mathbf{x}}_i(k\mid k)+\sum\limits_{s=1}^{l}\mathbf{A}_{ii}^{s-1}\,\mathbf{B}_{ii}\,\mathbf{u}_i(k+l-s\mid k)+\sum\limits_{s=1}^{l}\mathbf{A}_{ii}^{s-1}\,\hat{\mathbf{w}}_i(k+l-s\mid k-1),\\ \hat{\mathbf{y}}_i(k+l\mid k)=\mathbf{C}_{ii}\,\hat{\mathbf{x}}_i(k+l\mid k)+\hat{\mathbf{v}}_i(k+l\mid k-1).\end{cases} \tag{7}$$

Above all, the optimization problem for each subsystem-based-and-Local-cost optimization-based MPC in each control cycle can be concluded as:

Problem 3.1.

For each independent controller C_i, $i=1, 2, \cdots, n$, the unconstrained LCO-DMPC problem with prediction horizon P and control horizon M, $M<P$ at time k becomes the following optimization problem:

$$\min_{\Delta\mathrm{U}_i(k,M\mid k)}\bar{J}_i(k)=\sum_{l=1}^{P}\|\,\hat{\mathbf{y}}_i(k+l|k)-\mathbf{y}_i^d(k+l|k)\,\|_{\mathbf{Q}_i}^2+\sum_{l=1}^{M}\|\,\Delta\mathbf{u}_i(k+l-1|k)\,\|_{\mathbf{R}_i}^2, \tag{8}$$

for $i=1, 2, \cdots, m$, subject to constraints:

$$\hat{\mathbf{x}}_i(k+l \mid k) = \mathbf{A}_{ii}^l \hat{\mathbf{x}}_i(k \mid k) + \sum_{s=1}^{l} \mathbf{A}_{ii}^{s-1} \mathbf{B}_{ii} \mathbf{u}_i(k+l-s \mid k) + \sum_{s=1}^{l} \mathbf{A}_{ii}^{s-1} \hat{\mathbf{w}}_i(k+l-s \mid k-1), \quad (9)$$

$$\hat{\mathbf{y}}_i(k+l \mid k) = \mathbf{C}_{ii} \hat{\mathbf{x}}_i(k+l \mid k) + \hat{\mathbf{v}}_i(k+l \mid k-l), \quad (10)$$

where

$$\Delta \mathbf{U}_i(k,M|k) = \{\Delta \mathbf{u}_i(k \mid k), \cdots, \Delta \mathbf{u}_i(k+M-1 \mid k)\}.$$

Each controller C_i is composed of three parts: an optimizer, a state predictor and an interaction predictor. At time k, based on the exchanged information, the interaction predictor of MPC controller C_i estimates the future interaction sequence over the prediction horizon $\hat{w}_i(k+l-1 \mid k-1)$, $l=1, 2, \cdots, P$. Then, combining with the local state measurement of $x_i(k)$, problem 3.1 can be solved by the optimizer. The optimizer computes the optimal manipulated variable increments sequence $\Delta \mathbf{U}_i^*(k,M|k)$ over the control horizon. The first element of $\Delta U^*(k+M \mid k)$, $\Delta u_i(k \mid k)$ is selected and $u_i(k)=u_i(k-1)+\Delta u_i(k|k)$ is applied as control input to subsystem S_i. Finally, the state predictor computes an estimation of the future state trajectory over the prediction horizon by (9) and broadcasts it and the optimal control sequence $\Delta U^*(k+M|k)$ over the control horizon to its output neighbors S_j, $j \in \mathcal{P}_{-i}$. At time k, the interaction prediction part of each controller uses this information to estimate the interaction predictions $\hat{w}_i(k+l-1 \mid k)$ and the whole procedure is repeated.

It should be noticed that at Problem 3.1, the future interaction sequences are substituted by the estimation of the future interaction sequence $\hat{w}_i(k+l-s \mid k-1)$ and $\hat{v}_i(k+l \mid k-1)$ based on the information broadcasted at time $k-1$ from the agents $\mathcal{C}_j \in \mathcal{P}_{+i}$, because at time k, the predictions $\hat{w}_i(k+l-s \mid k)$ and $\hat{v}_i(k+l \mid k)$ are unknown for the controller C_i. That is why equations (9) and (10) in controller C_i have the current formation.

Unconstrained LCO-DMPC Algorithm

If the desired output $\mathbf{Y}_i^d(k+1,P \mid k)$ is provided, the LCO-DMPC algorithm for subsystem-based MPC in controller $\mathcal{C}_i$ at each time instant k is as follows.

Step 1 Communication and Interaction calculation

- Send $\mathbf{U}_i(k-1,M \mid k-1)$ and $\hat{\mathbf{X}}_i(k,P \mid k-1)$ to its downstream neighbors' controller $\mathcal{C}_j, j \in \mathcal{P}_{-i}$.
- Get the estimation of the future state trajectories $\hat{\mathbf{X}}_j(k,P \mid k-1)$ and control inputs $\mathbf{U}_j(k-1,M \mid k-1)$ from its upstream neighbors' controller $\mathcal{C}_j, j \in \mathcal{P}_{+i}$ through network information exchange.
- Set the desired trajectory $\mathbf{Y}^d(k+1,P \mid k)$ over the horizon P according to the MPC's configuration.
- Get the measurement of $\mathbf{x}_i(k)$ through field instruments or a designed observer.
- Build $\hat{\mathbf{X}}(k,P \mid k-1)$ and $\mathbf{U}(k,P \mid k)$ by combining the local state trajectory $\hat{\mathbf{X}}_i(k,P \mid k-1)$ and control input $\mathbf{U}(k, P \mid k)$ with the acquired upstream neighbors' information of $\hat{\mathbf{X}}_j(k,P \mid k-1)$ $\mathbf{U}_j(k-1,M \mid k-1)$, $j \in \mathcal{P}_{+i}$, and compute the corresponding predictions of the interactions:

$$\hat{\mathbf{W}}_i(k,P \mid k-1) = \tilde{\mathbf{A}}_i \hat{\mathbf{X}}(k,P \mid k-1) + \tilde{\mathbf{B}}_i \mathbf{U}(k-1,M \mid k-1),$$

$$\hat{\mathbf{V}}_i(k,P \mid k-1) = \hat{\mathbf{C}}_i \hat{\mathbf{X}}(k,P \mid k-1).$$

Step 2 Compute control law and apply it

- Compute the optimal control sequence. That is:

$$\mathbf{U}_i(k,M \mid k) = \Gamma_1' \mathbf{u}_i(k-1) + \bar{\Gamma}_i \bar{\mathbf{K}}_i [\mathbf{Y}_i^d(k+1,P \mid k) - \hat{\mathbf{Z}}_i(k+1,P \mid k)].$$

- Apply the first element $\mathbf{u}_i(k)$ of the optimal sequence $\mathbf{U}_i(k,M \mid k)$ as control input to physical system $\mathcal{S}_i$.

Step 3 Estimate the future state

- Compute the estimation of the future state trajectory of subsystem S_i over the horizon P by the following equation:

$$\hat{\mathbf{X}}_i(k+1,P \mid k) = \bar{\mathbf{S}}_i[\bar{\mathbf{A}}_i \hat{x}_i(k \mid k) + \bar{\mathbf{B}}_i \mathbf{U}_i(k,M \mid k) + \tilde{\mathbf{A}}_i \hat{\mathbf{X}}(k,P \mid k-1) + \tilde{\mathbf{B}}_i \mathbf{U}(k-1,M \mid k-1).$$

Step 4 Go to next time instant

At time $k+1$, let $k+1 \rightarrow k$, then go to Step 1 and repeat the algorithm.

According to the fault tolerant control [37], the LCO-DMPC control solution is able to manage also eventual subsystem faults. For example, for controller C_i, if a fault occurs and leads a structural or parametric change on model of subsystem S_i, then by model based techniques [38], controller S_i detects the occurred fault, and then determines the new configuration of LCO-DMPC and broadcasts the new configuration to its downstream neighbors controller C_i, $j \in \mathcal{P}_{-i}$. The controller C_i could switch to a new MPC policy according to the configuration and go on control its corresponding subsystem. Here, the local fault-detection system has to have the functions of fault detecting, fault type determining and automatic selecting configuration. It should also be able to inform the downstream neighbor controller of the new configuration and go on control its corresponding subsystem.

The stability condition of the entire closed-loop system is deduced by analyzing the entire closed-loop systems' dynamic matrix which could be specified on the basis of the closed-loop solution.

3.2 Distributed MPC strategy based on Nash optimality

For large-scale systems, to avoid the prohibitively high on-line computational demand and improve the global performance compared with the non-iterative LCO-DMPC, this section will introduce the DMPC control based on Nash optimality.

It is assumed that the behavior of the whole system is described by m subsystems and the nonlinear performance function L is decomposable in the distributed system. The local performance index for the i_{th} controller can be expressed as:

$$\min_{\Delta u_{i,M}(k|k)} J_i = \sum_{j=1}^{P} \mathrm{L}_i [\mathbf{y}_i(k+j\,|\,k)\Delta\,\mathbf{u}_{i,M}(k\mid k)](i=1,\cdots,m), \tag{11}$$

where L_i is the nonlinear function of $y_i(k+j\mid k)$, $\Delta u_{i,M}(k\mid k)$. It indicates the global performance index of the whole system is:

$$\min J = \sum_{i=1}^{m} J_i. \tag{12}$$

At time instant k, the future predictive output of the ith controller can be expressed as:

$$\mathbf{y}_i(k+j\,|\,k) = \mathrm{f}_i(\mathbf{y}_i(k),\Delta\,\mathbf{u}_{1,M}(k\mid k),\cdots,\Delta\,\mathbf{u}_{m,M}(k\mid k))(j=1,\cdots,P). \tag{13}$$

It can be concluded that the global performance index can be decomposed into a number of local performance indexes, but the output of each subsystem is still related to all the input variables due to the input coupling. Such distributed control problem with different goals can be resolved by means of Nash optimal concept [39]. In concrete, the group of control decisions $\mathbf{u}^N(t)=\{\mathbf{u}_1^N(t),\cdots,\mathbf{u}_m^N(t)\}$ is called the Nash optimal solution if for all u_i, the following relations are held:

$$J_i^*(\mathbf{u}_1^N,\cdots\mathbf{u}_i^N,\cdots,\mathbf{u}_m^N)\leq J_i(\mathbf{u}_1^N,\cdots,\mathbf{u}_{i-1}^N,\mathbf{u}_i,\mathbf{u}_{i+1}^N,\cdots,\mathbf{u}_m^N). \tag{14}$$

If the Nash optimal solution is adopted, no controller changes its control decision ui because it has achieved the locally optimal objective under the above-mentioned condition; otherwise, the local performance index Ji will degrade. Each controller optimizes its objective (local performance index) only using its own control decision, assuming that other controllers' Nash optimal solutions have been known. That is:

$$\min_{u_i} J_i\,|_{u_j^N(j\neq i)}. \tag{15}$$

Employing (15) to obtain the Nash optimal solution ui of the ith subsystem, it is necessary to know other subsystems Nash optimal solutions $\mathbf{u}_j^N(j\neq i)$, so that the whole system could arrive at Nash optimal equilibrium in this coupling decision process. By Nash optimal equilibrium the global optimization problem can be decomposed into a number of local optimization problems.

An iterative algorithm is developed on the basis of the previous work [40] to seek the Nash optimal solution of the whole system at each sampling time. Since the mutual communication and the information exchange are adequately taken into account, each controller can resolve local optimal problems provided that the other subsystem-based MPCs optimal solutions have been known. Then each subsystem-based MPC compares the newly computed optimal solution with the one obtained in last iteration and checks if the terminal condition is satisfied. If the algorithm is convergent, all the terminal conditions of the m agents will be satisfied, and the whole system will arrive at Nash equilibrium at this time. This Nash-optimization process will be repeated at next sampling time.

To avoid the prohibitively high on-line computational demand, the MPC is implemented in distributed scheme with the inexpensive controllers in the network environment. These controllers can co-operate and communicate with each other to achieve the objective of the whole system. Coupling effects among the agents are fully taken into account in this scheme, which is superior to other traditional decentralized control methods. The main advantage of this scheme is that the on-line optimization of a large-scale system can be converted to that of several small-scale systems. Thus, it can significantly reduce the computational complexity while ensuring satisfactory performance.

Step 1: At sampling time instant k, each controller makes initial estimation of the input variables and announces it to the other controllers, let the iterative index $l = 0$.

Step 2: Each controller resolves its optimal problem simultaneously to obtain its solution $\Delta \mathbf{u}_{i,M}^{*}(k)(i = 1,\cdots,m)$

$$\Delta \bar{\mathbf{u}}_{i,M}^{l}(k) = [\Delta \bar{\mathbf{u}}_{i}^{l}(k), \Delta \bar{\mathbf{u}}_{i}^{l}(k+1), \cdots, \Delta \bar{\mathbf{u}}_{i}^{l}(k+M-1)]^{\mathrm{T}} (i = 1,\cdots,m)$$

Step 3: Each controller checks if its terminal iteration condition is satisfied. That is, for the given error accuracy $\varepsilon_i (i = 1, \cdots, m)$, if there is:

$$\| \Delta \mathbf{u}_{i,M}^{(l+1)}(k) - \Delta \mathbf{u}_{i,M}^{(l)}(k) \| \leq \varepsilon_i \quad (i = 1,\cdots,m)$$

and all the terminal conditions are satisfied, then end the iteration and go to step 4; Otherwise, let $l = l+1, \Delta \mathbf{u}_{i,M}^{l}(k) = \Delta \mathbf{u}_{i,M}^{*}(k)(i = 1,\cdots,m)$, all controllers communicate to exchange this information, and take the latest solution to step 2.

Step 4: Compute the instant control law

$$\Delta \mathbf{u}_i(k) = [\mathbf{I} \quad 0 \quad 0 \quad \cdots \quad 0] \Delta \mathbf{u}_{i,M}^{*}(k)(i = 1,\cdots,m)$$

and take them as the controller outputs of each agent.

Step 5: Move horizon to the next sampling time. That is, $k+1 \rightarrow k$ and go to step 1 to repeat the above process.

3.3 Cooperative distributed MPC

As is introduced in the previous section, the optimization performance of the closed-loop system under the control of Distributed Predictive Control is usually not as good as that under the control of centralized Predictive Control, especially when the strong coupling exists among subsystems. As presented above, if the iterative algorithm is employed in solving each subsystem-based Predictive Control, in which, each subsystem controller communicates several times with its neighbors and solves the Quadratic Programing problem several times in each control period, it improves the global performance through minimizing the computational error, which refers to the difference between the input sequence calculated at previous iterative and the input sequence calculated in current computation. However, the research direction of the whole optimization problem is not the gradient of the entire cost function, and the optimal solution calculated by this method is Nash Optimality instead of the global optimality.

Is there any other strategy to improve the global performance of closed-loop system under the control of Distributed Predictive Control ?Du [40] and Venkat [41] proposed a strategy where each subsystem-based Predictive Control optimizes not only the cost function of the subsystem it corresponded to but also that of the whole system to improve the performance of entire closed-loop system. The advantage of improving the optimization performance of entire closed-loop system has been proved [10, 22, 31], and some applications are also presented to validate this strategy [22, 42, 43]. To introduce the concept more clearly, the unconstraint DPC [18, 31], both iterative and non-iterative algorithm, based on this coordination strategy are presented. In this strategy, each subsystem-based MPC should have access to the required information of all subsystems for calculating its optimal solution.

(1) Non-iterative cooperative distributed MPC: Assumptions:

- Controllers are synchronous, since the sampling interval is usually rather long compared with the computational time in process control;

- Communication channel introduces a delay of a single sampling time interval, since an instantaneous data transfer is not possible in real situations;
- Controllers communicate only once within a sampling time interval;
- Local states $x_i(k)$, $i=1, 2, \cdots, m$, are accessible.

Since the optimal control decision of S_i affects, or even destroys, the performance of other subsystems, the performance of other subsystems should be considered in finding the optimal solution of S_i. To improve the global performance of whole closed-loop system, the following so-called global performance index is adopted in each C_i, $i=1, 2, \cdots, m$:

$$\bar{J}_i(k) = \sum_{l=1}^{P} \| \hat{\mathbf{y}}^i(k+l|k) - \mathbf{y}^d(k+l|k) \|_{\mathbf{Q}}^2 + \sum_{l=1}^{M} \| \Delta \mathbf{u}_i(k+l-1|k) \|_{\mathbf{R}_i}^2, \tag{16}$$

where $Q=diag\{Q_1, Q_2, \cdots, Qm\}$. It should be noticed that $\Delta u_i(k+l+1|k)$ is excluded in the performance index, since it is independent of the future inputs sequence of S_i.

Since the state evolution of other subsystems is affected by $u_i(k)$ after one or several control periods, to improve the prediction precision, this influence is considered in C_i when predicting the future states of all subsystems. In addition, due to the unit delay introduced by the network, the information of other subsystems is available only after one sampling time interval. Therefore, in C_i, the states and outputs of all subsystems in l-step ahead are predicted by

$$\begin{cases} \hat{\mathbf{x}}^i(k+l+1|k) = \mathbf{A}^l \mathbf{L}_i \mathbf{x}(k) + \mathbf{A}^l \mathbf{L}'_i \hat{\mathbf{x}}(k|k-1) \\ \qquad + \sum_{s=1}^{l} \mathbf{A}^{s-1} \mathbf{B}_i \mathbf{u}_i(k+l|k) + \sum_{\substack{j\in\{1,\cdots,m\} \\ j\neq i}} \sum_{s=1}^{l} \mathbf{A}^{s-1} \mathbf{B}_j u_j(k+l|k-1), \\ \hat{\mathbf{y}}^i(k+l+1|k) = \mathbf{C} \hat{\mathbf{x}}^i(k+l+1|k), \end{cases} \tag{17}$$

where

$$\mathbf{L}_i = [\mathbf{0}_{n_{xi}\times\sum_{j=1}^{i-1} n_{xj}} \quad \mathbf{I}_{n_{xi}} \quad \mathbf{0}_{n_{xi}\times\sum_{j=i+1}^{m} n_{xj}}];$$
$$\mathbf{L}'_i = diag\{I_{\sum_{j=1}^{i-1} n_{xj}}, 0_{n_{xi}}, I_{\sum_{j=i+1}^{m} n_{xj}}\};$$
$$\mathbf{B}_i = [\mathbf{B}_{1i}^{\mathrm{T}} \quad \mathbf{B}_{2i}^{\mathrm{T}} \quad \cdots \quad \mathbf{B}_{mi}^{\mathrm{T}}]^{\mathrm{T}}.$$

It should be noticed that the input of this neighborhood model is still the input of S_i and the inputs and states of other subsystems are regarded as disturbances. The estimations of future states and outputs of all subsystems (except S_i) are only used in controller C_i, and these estimations are different from those estimated by the controller C_i itself.

On the basis of the closed-form solution, the closed-loop dynamics can be specified and the stability condition can be verified by analyzing the closed-loop dynamic matrix. In C-DMPC, the initial states and future control sequences of other subsystems at time k are substituted by the estimations calculated at time $k-1$. If there is disturbance, model mismatch or set point change, the future input sequences of subsystems calculated at time k are not equal to those calculated at time $k-1$, which induces estimation errors of future states between two optimization strategies. This affects the final performance of the closed-loop system. Although this difference exists, the optimization problem of C-DMPC is still very close to that of centralized MPC.

(2) Distributed Predictive Control based on Pareto Optimality: The main idea of this method is that: each subsystem-based MPC communicates with each other many times in a control period and computes the optimal control law through iteration; each subsystem-based MPC optimizes the cost of entire closed-loop system before improving the global performance of closed-loop system.

Step 1: At time k, each subsystem transmits the initial predictive value $y_{i,0}(k)$ to other subsystems and receives the initial predictive values $y_{j,0}(k)(j=1,2,\cdots,m,j\neq i)$ from other subsystems. Then it sends the estimator of the control law to other subsystems. Let the iteration number $l=0$.

Step 2: Each subsystem computes the optimal problem in parallel with last optimal control laws from other subsystems. The constraints are the same as mentioned above. We can get the optimal solution $\Delta\mathbf{u}_i^{l+1}(k)(i=1,2,\cdots,m)$ at the iteration.

Step 3: Check the convergence condition of all the subsystems. If the precision $\varepsilon_i(i=1,2,\cdots,m)$ of all the subsystems meets the condition

$$\|\Delta\mathbf{u}_i^{l+1}(k)+\Delta\mathbf{u}_i^{l}(k)\|\leq\varepsilon_i,$$

we can get $\Delta\mathbf{u}_i^{*}(k)=\Delta\mathbf{u}_i^{l+1}(k)(i=1,2,\cdots,m)$ and this iteration ends and it goes to step 4. Otherwise, let $\Delta\mathbf{u}_i^{l}(k)=\Delta\mathbf{u}_i^{l+1}(k)(i=1,2,\cdots,m)$, $l=l+1$ and go to step 2.

Step 4: Compute the control law at the time k

$$\Delta\mathbf{u}_i(k)=[I0\cdots0]\Delta\mathbf{u}_{i,M}^{*}(k)i=1,2,\cdots,m$$

Step 5: Let $k+1\rightarrow k$ and go to the step 1.

During the optimal control time, each subsystem can get a local MPC control law without considering the coupling relationship between subsystems. The local MPC control law will be taken as the iteration initial value.

The on-line optimization of a large-scale system can be converted to that of several small-scale systems. It can thus significantly reduce the computational complexity while keeping satisfactory performance.

3.4 The networked Distributed MPC

The major advantage of distributed model predictive control is that it has the characteristics of good flexibility and error tolerance. This characteristic is based on the fact that the controllers relevant are independent. It means that the number of systems that each subsystem based MPC communicates with decreased, which will improve the flexibility and the ability of error-tolerance of the whole closed-loop control system. In addition, in some fields or processes the global information is unavailable to controllers (e.g. in multi intelligent vehicle system) for the management or the system scale reasons. Thus, designing a DMPC which could significantly improve the global performance of the closed-loop system with limited information structure constraints is valuable.

We will propose a coordination strategy which could improve the global performance using appropriate network resources, where the optimization objective of each subsystem-based MPC considers the performance of corresponding local subsystems and its directly impacted subsystems. In online optimization problem, each local controller takes into account not only the impacts coming from its neighbors but also the impacts applied to its neighbors for improving global performance.

For the non-iterative algorithm, the closed-loop stability analysis is also provided for guiding local MPCs tuning. Moreover, the performance of closed-loop system using proposed distributed MPC is analyzed and the application to accelerated cooling and controlled (ACC) process is presented to validate the efficiency of this method. For the iterative algorithm where each subsystem based MPC exchanges information several times when it solves its local optimization problem, the optimality of the iteration based networked MPC algorithm is analyzed and the nominal stability is derived for distributed control systems without the control and output constraints.

(1) Non-iterative Networked DPC: Since the state evolution of downstream neighbors of subsystem S_i is affected by the control decision of subsystem S_i, the performance of these neighbors may be destroyed by improper control decisions in some cases. To solve this problem, the so-called Neighbourhood optimization [23, 26] is adopted and the performance index is expressed as:

$$\begin{aligned}\bar{J}_i(k)&=\sum_{j\in N_i^{out}}J_i(k)\\&=\sum_{j\in N_i^{out}}\Big[\sum_{l=1}^{P}\|\hat{\mathbf{y}}_j(k+l|k)-\mathbf{y}_j^d(k+l|k)\|_{\mathbf{Q}_j}^2+\sum_{l=1}^{M}\|\Delta\mathbf{u}_j(k+l-1|k)\|_{\mathbf{R}_j}^2\Big].\end{aligned}\tag{18}$$

Since $\Delta\mathbf{u}_j(k+l-1|k)(j\in\mathcal{P}_{-i},j\neq i,l=1,\cdots,M)$ is unknown and independent of the control decision of

$\mathcal{S}_i$, $\Delta u_j(k+l-1|k-1)$ is used to approximate $\Delta \mathbf{u}_j(k+l-1|k)$. Then, the performance index becomes

$$\begin{aligned}\bar{J}_i(k) &= \sum_{j\in\mathcal{P}_{-i}}\sum_{l=1}^{P}\|\hat{\mathbf{y}}_j(k+l|k)-y_j^d(k+l|k)\|_{\mathbf{Q}_j}^2+\sum_{l=1}^{M}\|\Delta\mathbf{u}_i(k+l-1|k)\|_{\mathbf{R}_i}^2\\ &+\sum_{j\in\mathcal{P}_{-i},j\neq i}\sum_{l=1}^{M}\|\Delta\mathbf{u}_i(k+l-1|k-1)\|_{\mathbf{R}_j}^2\\ &=\sum_{j\in\mathcal{P}_{-i}}\sum_{l=1}^{P}\|\hat{\mathbf{y}}_j(k+l|k)-\mathbf{y}_j^d(k+l|k)\|_{\mathbf{Q}_j}^2+\sum_{l=1}^{M}\|\Delta\mathbf{u}_i(k+l-1|k)\|_{\mathbf{R}_i}^2+\text{Constant}.\end{aligned} \tag{19}$$

The optimization index $\bar{J}_i(k)$ considers not only the performance of subsystem S_i but also that of the downstream neighbors of S_i. The impacts of the control decision of S_i to $\mathcal{S}_j\in\mathcal{P}_{-i}$ are fully considered in the neighborhood optimization, and therefore the global performance improving is guaranteed. It should be noticed that the global performance may be further improved by using the optimization objective in each subsystem, but it requires a high quality and complicated network communication and introduces more complex computation.

In fact, after several control periods, the control decision $\Delta U_i(k, M|k)$ affects not only the downstream neighbors of S_i but also other subsystems (e. g. the downstream neighbors of the downstream neighbors of S_i). Here, the interactions with other subsystems except downstream neighbors are neglected. If there is enough network band-width for employing iterative algorithm, these interactions can also be taken into account.

It should be noticed that each controller only communicates with its neighbors and its neighbors' neighbors in ND-MPC. Moreover, if each controller communicates with its neighbors twice within a sampling time interval, the information of its neighbors' neighbors can be obtained from its neighbors. That means only the information exchanging among neighborhood is required by this method. Thus, if one subsystem fails, the other subsystem unrelated to S_i can run normally. The communication loads related to S_i are that S_i gets its future states to its neighbors and sends its neighbors' states and inputs to its neighbors.

(2) Networked DMPC with Iterative Algorithm: As was mentioned, the closed-loop system will achieve Nash optimality if the iterative algorithm is employed in the local cost optimization based DMPC and the closed-loop system could obtain Pareto Optimality if the iterative algorithm is employed in the global cost optimization based DMPC. The iteration could indeed improve the global performance of closed-loop system in the distributed control framework.

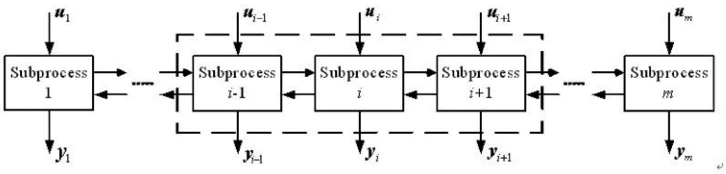


Figure 2 Diagram of a serially connected process.

Step 1. Initialization and communication: At the sampling time instant k , each subsystem exchanges $\hat{\mathbf{x}}_i(k)$ with its neighbors, makes the initial estimate of its local optimal control decision and transmits it to its neighbors by a communicator. Let the iterative index $l=0$:

$$\Delta\mathbf{U}_{i,M}^{(l)}(k)=\Delta\hat{\mathbf{U}}_{i,M}(k)(i=1,\cdots,m)$$

Step 2. Subsystem optimization: Each subsystem resolves its local optimization problem described in

$$\min_{\Delta U_{i,M}(k)}\bar{J}_i(k)\,|_{\Delta\mathbf{U}_{j,M}^{(l)}(k)(j\in\mathbb{N}_i,j\neq i)}$$

simultaneously to derive its control decision $\Delta\mathbf{U}_{i,M}^{(l+1)}(k)$.

Step 3. Checking and updating: Each subsystem checks if its terminal iteration condition is satisfied. That is, for the given error accuracy $\varepsilon_i \in \mathbb{R}$, $(i=1,\cdots,m)$, if there exists

$$\| \Delta \mathbf{U}_{i,M}^{(l+1)}(k) - \Delta \mathbf{U}_{i,M}^{(l)}(k) \| \leq \varepsilon_i,\ (i=1,\cdots,m)$$

and all the terminal conditions are satisfied at iteration l^* , then end the iteration, set the local optimal control decision for each subsystem $\Delta \mathbf{U}_{i,M}^{*}(k) = \Delta \mathbf{U}_{i,M}^{(l^*)}(k)$, and go to Step 4; otherwise, let $l = l+1$, each subsystem communicates to exchange the new information $\Delta U_{i,M}^{(l)}(k)$ with its neighbors, and go to Step 2.

Step 4. Assignment and implementation: Compute the instant control law

$$\Delta \mathbf{u}_i^*(k) = \begin{bmatrix} \mathbf{I}_{n_{ui}} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \Delta \mathbf{U}_{i,M}^{*}(k), (i=1,\cdots,m)$$

and apply $u_i^*(k) = \Delta u_i^*(k) + u_i^*(k-1)$ to each subsystem.

Step 5. Reassigning the initial estimation: Set the initial estimation of the local optimal control decision for the next sampling time

$$\Delta \hat{U}_{i,M}(k+1) = \Delta U_{i,M}^{*}(k)(i=1,\cdots,m)$$

Step 6. Receding horizon: Move horizon to the next sampling time. That is, $k+1 \rightarrow k$, go to Step 1, and repeat the above steps.

The on-line optimization of serially connected large-scale systems can be converted to that of several small-scale systems via distributed computation. It can thus significantly reduce the computational complexity. Meanwhile, information exchange among neighboring subsystems in a distributed structure via communication can improve control performance, which is superior to traditional decentralized MPC methods.

Under the network environment, the capacity of the communication network is assumed to be sufficient for each subsystem to obtain information from its neighbors, so it is possible for each subsystem to exchange information several times when it solves its local optimization problem at the sampling time instant. Furthermore, when the convergent condition is satisfied, the solution to the local optimization problems collectively will be the global optimal control decision of the whole system. That is, the coordinated distributed computations solve an equivalent centralized MPC problem.

It has been noticed that the convergence of N-MPC is local. That is to say, whether the distributed computation is convergent is only concerned at the current sampling time instant. The stability analysis in this section is global. That is to say, the convergence of the distributed computation and stability for distributed control systems are considered during the whole receding horizon.

4 The design methods of the stabilizing distributed MPCS with constraints

Control design that takes state and/or input constraints into account, whether under the MPC framework or not, is an important and challenging problem. Many methods have been reported [44—46]. Under the MPC framework, closed-loop stability is ensured by judiciously integrating designs of the terminal cost, the terminal constraint set and the local controllers [44]. In DMPC, the future state sequences of upstream neighbors, which are calculated based on the solution in the previous time instant, may not be equal to the predictive states calculated by the corresponding subsystem at the current time instant, and the errors between them are hard to estimate. In addition, in the presence of constraints, the feasibility of each subsystem based MPC cannot be guaranteed. The remaining part of the optimal control sequence calculated at the previous time instant may not be a feasible solution at the current time instant. It is difficult to construct a feasible solution in the current time instant. All these make it difficult to design a stabilizing LCO-DMPC that takes constraints into consideration.

A stabilized LCO-DMPC algorithm is developed, which uses constraints to limit the error between the future state sequences (or called presumed sequences) of upstream neighbors, which are calculated based on the solution in the previous time instant, and the predictive states calculated by the corresponding subsystem in the current time instant. Then the stability is ensured by judiciously integrating designs of the bound of the error between presumed state sequence, predictive state sequence, the terminal cost, the constraint set and the local controllers.

4.1 The local cost optimization based distributed MPC based on dual mode scheme

Each local controller minimizes its own subsystems cost and uses the state prediction of the previous time instant to approximate the state sequence at the current time instant in computing the optimal solution. If iterative algorithm is employed, the Nash Optimality of closed-loop system can be achieved.

Each subsystem-based MPC minimizes the cost function of its corresponding subsystem. More specifically, the performance index is defined as

$$J_i(k) = \| \mathbf{x}_i^p(k+N \mid k) \|_{P_i}^2 + \sum_{s=0}^{N-1} (\| \mathbf{x}_i^{\mathbf{p}}(k+s \mid k) \|_{\mathbf{Q}_i}^2 + \| \mathbf{u}_i(k+s \mid k) \|_{\mathbf{R}_i}^2) \tag{20}$$

where $\mathbf{Q}_i = \mathbf{Q}_i^{\mathrm{T}} > 0$, $\mathbf{R}_i = \mathbf{R}_i^{\mathrm{T}} > 0$ and $\mathbf{P}_i = \mathbf{P}_i^{\mathrm{T}} > 0$. The matrix $\mathbf{P}_i$ is chosen to satisfy the Lyapunov equation $\mathbf{A}_{di}^{\mathrm{T}} \mathbf{P}_i \mathbf{A}_{di} - \mathbf{P}_i = -\hat{\mathbf{Q}}_i$, where $\hat{\mathbf{Q}}_i = \mathbf{Q}_i + \mathbf{K}_i^{\mathrm{T}} \mathbf{R}_i \mathbf{K}_i$.

In order to guarantee the stability of DMPC, we consider the following constraints:

$$\sum_{l=1}^{s} \alpha_{s-l} \,||\, \mathbf{x}_i^p(k+l \mid k) - \hat{\mathbf{x}}_i(k+l \mid k) \,||_2 \le \frac{\xi\kappa\varepsilon}{2\sqrt{mm_1}}, s = 1,2,\cdots,N-1, \tag{21}$$

$$\| \mathbf{x}_i^p(k+N \mid k) - \hat{\mathbf{x}}_i(k+N \mid k) \|_{P_i} \le \frac{\kappa\varepsilon}{2\sqrt{m}}, \tag{22}$$

$$\| \mathbf{x}_i^p(k+s \mid k) \|_{P_i} \le \| \mathbf{x}_i^{\mathrm{f}}(k+s \mid k) \|_{\mathbf{P}_i} + \frac{\varepsilon}{\mu N \sqrt{m}}, s = 1,2,\cdots,N, \tag{23}$$

$$\mathbf{u}_i^{\mathrm{p}}(k+s \mid k) \in \mathcal{U}_i, s = 0,1,\cdots,N-1, \tag{24}$$

$$\mathbf{x}_i^{\mathrm{p}}(k+N \mid k) \in \Omega_i(\varepsilon/2). \tag{25}$$

In the constraints above,

$$m_2 = \max_{i \in \mathcal{P}} \{\text{number of elements in } \mathcal{P}^{+i}\},$$

$$\alpha_l = \max_{i \in \mathcal{P}} \max_{j \in \mathcal{P}_i} \{\lambda_{\max}^{\frac{1}{2}}((\mathbf{A}_{ii}^l \mathbf{A}_{ij})^{\mathrm{T}} \mathbf{P}_j \mathbf{A}_{ii}^l \mathbf{A}_{ij})\}, \ l = 0,1,\cdots,N-1.$$

The constants $0 < \kappa < 1$ and $0 < \xi \le 1$ are design parameters whose values will be chosen in the sequel.

4.2 Cooperative distributed predictive control with constraints

Each subsystem-based MPC optimizes the cost of overall system to improve the global performance. In computing the optimal solution, it also uses the state prediction of the previous time instant to approximate the state sequence at the current time instant. This strategy could achieve a good global performance in some cases, but it reduces the flexibility and increases the communication load. We call it global cost optimization based DMPC here, and the Pareto Optimality of the closed-loop system is obtained by this method.

The consistency constraints, which limit the error between the optimal inputs sequence calculated at the previous time instant, referred to as the presumed inputs, and the optimal inputs sequence calculated at the current time instant to a prescribed bound, are designed and included in the optimization problem of each subsystem-based MPC. Moreover, a dual mode predictive control [44, 47] strategy is adopted. These consistency constraints and the dual mode strategy guarantee that the remaining part of the solution at the previous time instant is a feasible solution if there is a feasible solution at initial time instant. They also guarantee the asymptotical stability of the closed-loop system.

$$\sum_{h=0}^{l} \beta_{l-h} \| \mathbf{u}_i(k+h \mid k) - \mathbf{u}_i(k+h \mid k-1) \|_2 \le \frac{\gamma\kappa\alpha e}{m-1}, l = 1,2,\cdots,N-1; \tag{26}$$

$$\mathbf{u}_i(k+l-1 \mid k) \in \mathcal{U}_i, l = 0,1,\cdots,N-1; \tag{27}$$

$$\hat{\mathbf{x}}(k+N \mid k,i) \in \Omega(\alpha\varepsilon). \tag{28}$$

In the constraints above,

$$\beta_l = \max_{i\in P}(\lambda_{\max}((\mathbf{A}^l \bar{\mathbf{B}}_i)^T \mathbf{P} \mathbf{A}^l \bar{\mathbf{B}}_i)^{\frac{1}{2}}), l = 0,1,\cdots,N-1.$$

$$\lambda_{\max}(\sqrt{\mathbf{A}_c^T \mathbf{A}_c}) \leq 1-\kappa.\ 0<1-\kappa<1.$$

The constant $0<\kappa<1$, $0<\alpha<0.5$ and $\gamma>0$ are design parameters whose value will be chosen in the sequel.

Equation (26) is referred to as the consistency constraints, which requires that all predictive manipulated variables remain close to the presumed sequence. It is a key equation in proving that $x^f_{j,i}$ is a feasible state sequence at each updating.

Note that the terminal constraint in each optimal control problem is $\Omega(\alpha\varepsilon)$, $0<\alpha<0.5$. In the analysis presented in the next section, it is shown that tightening the terminal set in this way is required to guarantee the feasibility properties.

A stabilizing distributed implementation of MPC is developed for dynamically coupled spatially distributed systems subject to decoupled input constraints. Each subsystem-based MPC considers the performance of all subsystems and communicates with each other only once at a sampling time.

4.3 A networked distributed predictive control with inputs and information constraints

In an effort to achieve a trade-off between the global performance of the entire system and the computational burden, an intuitively appealing strategy is provided, where each subsystem-based MPC only considers the cost of its own subsystem and that of the subsystems it directly impacts on.

Under the DMPC framework, Dunbar [28] provided a design for nonlinear continuous systems, which used constraints to limit the error between the future state sequences (or called presumed sequences) of upstream neighbors, which were calculated based on the solution in the previous time instant, and the predictive states calculated by the corresponding subsystem in the current time instant. Then the stability was ensured by judiciously integrating designs of the bound of the error between presumed state sequence, predictive state sequence [28], the terminal cost, the constraint set and the local controllers [47]. Farina [29] gave another design for linear system, which used a fixed reference trajectory with a moving widow to substitute the presumed state/input of upstream neighbors used by Dunbar [28]. Both methods are designed for DMPC in which each subsystem-based MPC optimizes the cost of the corresponding subsystem itself. As for the DMPC which uses the global cost function, some convergence conditions are deduced if using iterative algorithms. Then the distributed problems can be reformulated into a centralized problem, and the stabilizing DMPC can be designed with a similar method of centralized MPC. For the coordination strategy used here, there is no global model that can be used. And except that there are errors between the presumed state/input sequences and predictive state sequences of upstream neighbors, the predictive state sequences of downstream neighbors calculated by current subsystem may not equal those calculated by the downstream neighbors themselves and these error are hard to be estimated. In the presence of constraints, the remaining part of the optimal control sequence calculated at the previous time instant may not be a feasible solution at the current time instant. All these make it difficult to design a stabilizing N-DMPC that takes constraints into consideration.

Considering subsystem $\mathcal{S}_i$, let $\varepsilon>0$ and the update time be $k\geq 1$, given $\mathbf{x}_i(k)$, $\mathbf{x}_{\bar{i}}(k)$ and $\hat{\mathbf{x}}_{\bar{i}}(k+s \mid k)$, $s=1,2,\cdots,N$, and $\hat{\mathbf{u}}_{\bar{i}}(k+s \mid k)$, $s=0,2,\cdots,N-1$, find the control sequence $\mathbf{u}_i^P(k+s \mid k):\{0,1,\cdots,N-1\} \rightarrow \mathcal{U}_i$ that minimizes

$$\bar{J}_i(k) = \sum_{j\in\mathcal{P}_i} \| \mathbf{x}^p_{j,i}(k+N|k) \|^2_{P_j} + \sum_{s=0}^{N-1}\Big(\sum_{j\in\mathcal{P}_i} \| \mathbf{x}^p_{j,i}(k+s|k) \|^2_{\mathbf{Q}_j} + \| \mathbf{u}^p_i(k+s|k) \|^2_{\mathbf{R}_i}\Big), \tag{29}$$

subject to the following constraints:

$$\sum_{l=0}^{s}\beta_{s-l}\ ||\ \mathbf{u}_i^{p}(k+l\mid k)-\hat{\mathbf{u}}_i(k+l\mid k)\ ||_2\leq\frac{(1-\xi)\kappa\varepsilon}{2\sqrt{mm_1}},s=1,2,\cdots,N-1,\tag{30}$$

$$\sum_{l=1}^{s}\alpha_{s-l}\ ||\ \mathbf{x}_i^{p}(k+l\mid k)-\hat{\mathbf{x}}_i(k+l\mid k)\ ||_2\leq\frac{\xi\kappa\varepsilon}{2\sqrt{mm_2}},s=1,2,\cdots,N-1,\tag{31}$$

$$\|\ \mathbf{x}_i^{p}(k+N\mid k)-\hat{\mathbf{x}}_i(k+N\mid k)\ \|_{P_i}\leq\frac{\kappa\varepsilon}{2\sqrt{m}},\tag{32}$$

$$\|\ \mathbf{x}_i^{p}(k+s\mid k)\ \|_{P_i}\leq\|\ \mathbf{x}_i^{f}(k+s\mid k)\ \|_{\mathbf{P}_i}+\frac{\varepsilon}{\mu N\sqrt{m}},s=1,2,\cdots,N,\tag{33}$$

$$\mathbf{u}_i^{p}(k+s\mid k)\in\mathcal{U}_i,s=0,1,\cdots,N-1,\tag{34}$$

$$\mathbf{x}_{j,i}^{p}(k+N\mid k)\in\Omega_j(\varepsilon/2),j\in\mathcal{P}_i.\tag{35}$$

In the constraints above,

$$m_1=\max_{i\in\mathcal{P}}\{\text{number of elements in }\mathcal{P}_{-i}\},$$

$$m_2=\max_{i\in\mathcal{P}}\{\text{number of elements in }\mathcal{P}_{\bar{i}}\},$$

$$\alpha_l=\max_{i\in\mathcal{P}}\max_{j\in\mathcal{P}_i}\{\lambda_{\max}^{\frac{1}{2}}((\mathbf{L}_{j,i}\ \bar{\mathbf{A}}_i^{l}\ \widetilde{\mathbf{A}}_i)^{\mathrm{T}}\ \mathbf{P}_j\ \mathbf{L}_{j,i}\ \bar{\mathbf{A}}_i^{l}\ \widetilde{\mathbf{A}}_i)\},$$

$$\beta_l=\max_{i\in\mathcal{P}}\max_{j\in\mathcal{P}_i}\{\lambda_{\max}^{\frac{1}{2}}((\mathbf{L}_{j,i}\ \bar{\mathbf{A}}_i^{l}\ \bar{\mathbf{B}}_{\bar{i}})^{\mathrm{T}}\ \mathbf{P}_j\ \mathbf{L}_{j,i}\ \bar{\mathbf{A}}_i^{l}\ \bar{\mathbf{B}}_{\bar{i}})\},$$

where

$$\mathbf{L}_{i,j}=\begin{bmatrix}\mathbf{0}_{n_{xj}\times\sum_{m_{l,i}<m_{j,i}}n_{xl}} & \mathbf{I}_{n_{xj}} & \mathbf{0}_{n_{xj}\times\sum_{m_{h,i}>m_{j,i}}n_{xh}}\end{bmatrix}$$

and $\mathcal{S}_j$, $\mathcal{S}_l$ and $\mathcal{S}_h$ are respectively the $(m_{j,i}-1)^{\text{th}}$, $(m_{l,i}-1)^{\text{th}}$ and $(m_{h,i}-1)^{\text{th}}$ subsystem in the downstream region of $\mathcal{S}_i$. Finally, the constants $0<\kappa<1$ and $0<\xi\leq 1$ are design parameters whose values will be chosen in the sequel.

Both the control decision and performance of the closed-loop system using N-DMPC are very close to those using centralized MPC. Furthermore, there is less computation demand using the N-DMPC than using centralized MPC. Thus, the N-DMPC is an effective method which could guarantee global performance improvement with higher computational speed and less communication burden. Compared with the method proposed in the last subsection, both the optimization index and the consistent constraints are different. In the optimal problem, the constraints (30) are necessary since the estimation error cannot be expressed by the states sequence. In addition, the terminal constraint should bound both the final states of corresponding subsystem and that of the subsystems it directly impacted on.

It should be noticed that, with the increasing of the number of the downstream neighbors each subsystem-based controller covers, the cost consumed by communication among subsystems will become even higher, and the network connectivity of the entire system will become more complex. When the time consumed by communicating becomes so much that it cannot be ignored compared with the control period, the performance of the entire system will be negatively affected more or less. And the increasing of network connectivity will inevitably violate the error tolerance capability of the entire control system. This is undesired in the distributed control framework. Thus, the number of the downstream neighbors of each subsystem should not be too big when the network bandwidth is limited or not big enough.

It should also be noticed that a general mathematical formulation is adopted in the N-DMPC algorithm and its analysis. The N-DMPC and the resulting analysis can be used for any coordination policy mentioned in Section II with a redefinition of $\mathcal{P}_i$. Thus it provides a unified framework for the DMPCs which adopts the cost function based coordination strategies.

5 Simulation example

5.1 Numerical Validation-Accelerated cooling process test rig

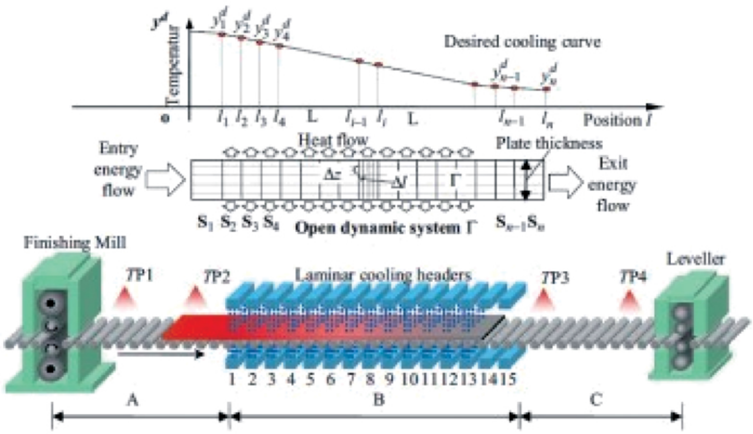


Figure 3 ACC process for the middle & heavy plate.

ACC process, simplified in Figure 3, is used to cool a metal plate from initial temperature around 750—800℃ down to the lowest temperature in the range of 450—560℃. A constant cooling curve of the plate is required in ACC, which helps a lot to strongly improve the mechanical characteristics of the corresponding products. The cooling area is partitioned into three sections: air cooling section, water cooling section and re-reddening section, labelled A, B and C, respectively. Fifteen cooling header units are uniformly spaced along section B. The number of cooling header units in operation (N), and the water flux of each cooling unit (F) can be adjusted separately. The temperature drop is caused by the heat radiation in sections A and C, and caused by both radiation and water cooling in section B [48]. Four pyrometers are located in the positions of 13.5, 58.6, 89 and 109.5 m, respectively. The temperatures of the plate inside the cooling section are measured by soft-sensors.

As for ACC process, the proposed N-DMPC is adopted in this work. As shown in Figure 4, each subsystem is controlled by a local MPC. As for the subsystems in which the corresponding cooling water header unit is closed, the local MPC is substituted with a predictor. The predictor estimates the future states of corresponding subsystem and broadcasts the estimations to its neighbors [49].

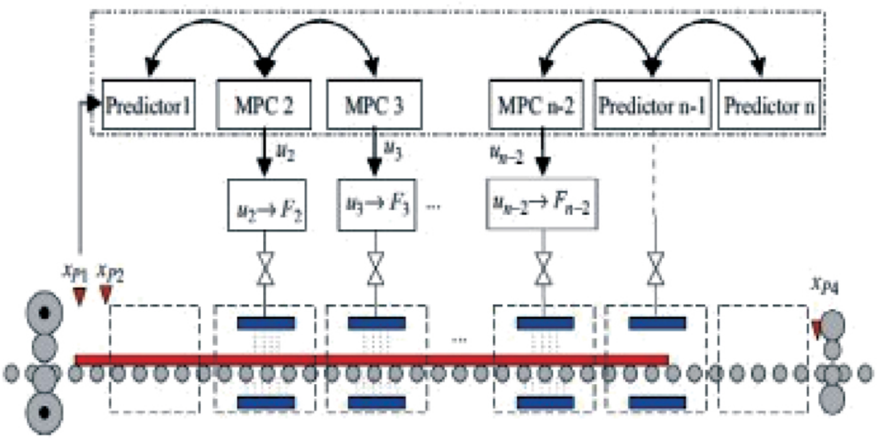


Figure 4 Control strategy of ACC.

5.2 Numerical Validation-Operation optimization of multi-type cooling source system based on distributed model predictive control

This section proposes a hierarchical distributed MPC strategy, which builds an economic model of the electric refrigerator and ices storage tank, and gets steady power states and optimal set point of each electric refrigerator and ice storage tank under optimal conditions by using mixed integer programming. Then it uses DMPC to make sure that each electric refrigerator and ice storage tank can track the upper optimal set point as soon as possible once the total power can track predictive load.

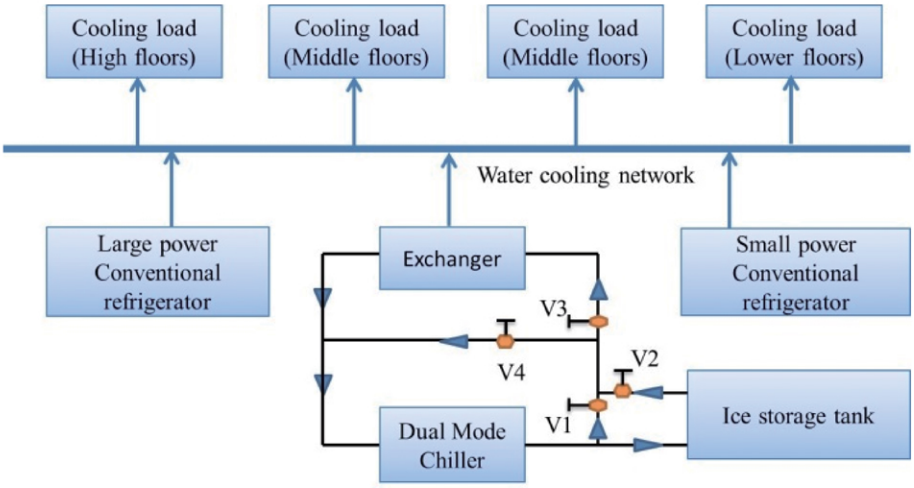


Figure 5 Structure of joint cooling system.

A typical multi-type cooling system is shown in Figure 5. The valves between the exchangers, ice storage and the dual mode electric refrigerator are used to control the switch of the working mode. Figure 6 is the whole control strategy structure of joint cooling system, where DR refers to the dual mode refrigerator and CR refers to conversational refrigerator. The single-pole double throw switch in the dynamic optimization level is used to choose the air-mode electric refrigerator controlled by the DMPC, and then the disabled electric refrigerator or ice-mode electric refrigerator is neglected. The virtual network is the data transmitting channel for MPC subsystems.

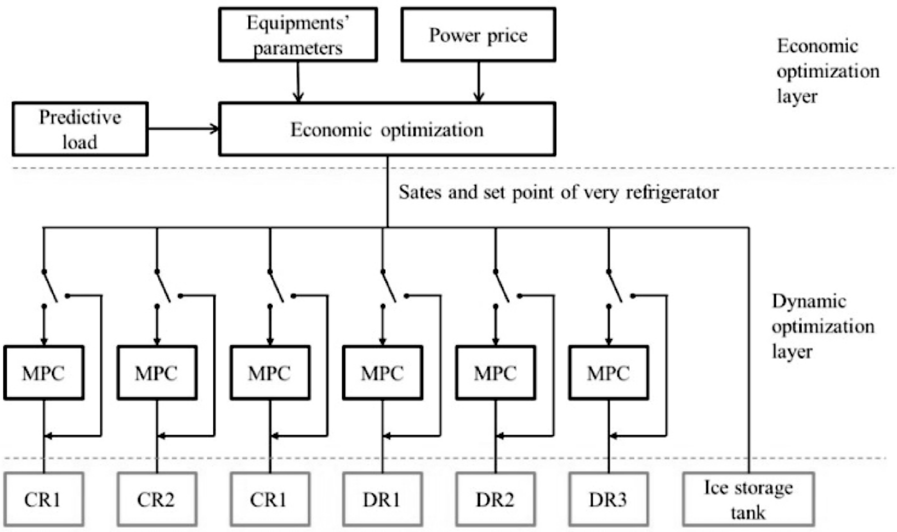


Figure 6 Control strategy of joint cooling system.

Above all, for the MPC controller of the th subsystem the optimal problem form is:

$$\min_{U_s} \quad J_s = Q\sum_{i=1}^{P} \left(r_s(k+i) - \hat{p}_s(k+i \mid k)\right)^2 + R\sum_{i=1}^{P} \left(r(k+i) - \hat{p}_s(k+i \mid k) - \sum_{\substack{j=1 \\ j\neq s}}^{N} \hat{p}_j(k+i \mid k)\right)^2,$$

$$\begin{aligned} s.t. \quad & \hat{\mathbf{x}}_s(k+i \mid k) = \mathbf{A}_s{}^{i}\,\mathbf{x}_s(k) + \sum_{j=1}^{i}(\mathbf{A}_s{}^{i-j}\,\mathbf{B}_s u_s(k+j-1)), \\ & \hat{p}_s(k+i \mid k) = \mathbf{C}_s\,\hat{\mathbf{x}}_s(k+i \mid k), \\ & u_{s,\min} \le u_s(k+i) \le u_{s,\max}, \\ & p_{s,\min} \le \hat{p}_s(k+i \mid k) \le p_{s,\max}, \\ & i = 1, 2, \cdots, P. \end{aligned}$$

6 Conclusion

In this paper, some significant theory and application results of DMPC were reviewed. From the distributed systems with constraints or not, different algorithms to handle related conditions were introduced. In the end, two examples were listed to show the feasibility of proposed algorithms. Moreover, the proposed algorithms have advantages and disadvantages; it needs more research on the DMPC to pursue better performance.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61233004, 61221003, 61374109, 61304078 & 61473184), the National Basic Research Program of China (973 Program) (Grant Nos. 2013CB035500 863 & 2015AA043102), and the Higher Education Research Fund for the Doctoral Program of China (Grant Nos. 20120073130006).

References

[1] Baheti R, Gill H, Cyber-physical systems. The impact of control technology, 2011: 161—166.

[2] Park KJ, Zheng R, Liu X. Cyber-physical systems: Milestones and research challenges. Computer Communications, 2012, 36(1): 1—7.

[3] Scattolini R. Architectures for distributed and hierarchical model predictive control-a review. Journal of Process Control, 2009, 19(5): 723—731.

[4] Christofides PD, Scattolini R, Muñoz de la Penñ a D, Liu J. Distributed model predictive control: A tutorial review and future research directions. Computers & Chemical Engineering, 2013, 51: 21—41.

[5] Li S, Zhang Y, Zhu Q. Nash-optimization enhanced distributed model predictive control applied to the shell benchmark problem. Information Sciences, 2005, 170(2): 329—349.

[6] Al-Gherwi W, Budman H, Elkamel A. Selection of control structure for distributed model predictive control in the presence of model errors. Journal of Process Control, 2010, 20(3): 270 — 284.

[7] Alvarado I, Limon D, de la Pena DM, Maestre J, Ridao M, Scheu H, Marquardt W, Negenborn R, De Schutter B, Valencia F, et al. A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark. Journal of Process Control, 2011, 21(5): 800—815.

[8] Stewart BT, Wright SJ, Rawlings JB. Cooperative distributed model predictive control for nonlinear systems. Journal of Process Control, 2011, 21(5): 698—704.

[9] del Real AJ, Arce A, Bordons C. An integrated framework for distributed model predictive control of large-scale power networks. IEEE Transactions on Industrial Informatics, 2014, 10(1): 197—209.

[10] Stewart BT, Venkat AN, Rawlings JB, Wright SJ, Pannocchia G. Cooperative distributed model predictive control. Systems & Control Letters, 2010, 59(8): 460—469.

[11] Camponogara E, Jia D, Krogh BH, Talukdar S. Distributed model predictive control. IEEE Control Systems, 2002, 22(1): 44—52.

[12] Leirens S, Zamora C, Negenborn R, De Schutter B. Coordination in urban water supply networks using distributed model predictive control. Proceedings of IEEE American Control Conference (ACC) 2010, 2010: 3957—3962.

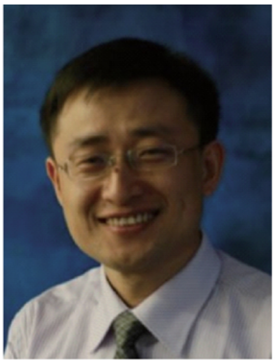
[13] D Moroşan-P, Bourdais R, Dumur D, Buisson J. Building temperature regulation using a distributed model predictive control. Energy and

Buildings, 2010, 42(9): 1445—1452.
[14] Giovanini L. Game approach to distributed model predictive control. IET Control Theory & Applications, 2011, 5(15): 1729—1739.
[15] Al-Gherwi W, Budman H, Elkamel A. A robust distributed model predictive control algorithm. Journal of Process Control, 2011, 21(8): 1127—1137.
[16] Scheu H, Marquardt W. Sensitivity-based coordination in distributed model predictive control. Journal of Process Control, 2011, 21(5): 715—728.
[17] Ferramosca A, Limon D, Alvarado I, Camacho EF. Cooperative distributed MPC for tracking. Automatica, 2013, 49(4): 906—914.
[18] Zheng Y, Li S, Qiu H. Networked coordination-based distributed model predictive control for large-scale system. IEEE Transactions on Control Systems Technology, 2013, 21(3): 991—998.
[19] Richalet J. Industrial applications of model based predictive control. Automatica, 1993, 29(5): 1251—1274.
[20] Zheng Y, Li S, Li N. Distributed model predictive control over network information exchange for large-scale systems. Control engineering practice, 2011, 19(7): 757—769.
[21] Vaccarini M, Longhi S, Katebi MR. Unconstrained networked decentralized model predictive control. Journal of Process Control, 2009, 19(2): 328—339.
[22] Venkat AN, Hiskens IA, Rawlings JB, Wright SJ. Distributed MPC strategies with application to power system automatic generation control. IEEE Transactions on Control Systems Technology, 2008, 16(6): 1192—1206.
[23] Zheng Y, Li S, Wang X. Distributed model predictive control for plant-wide hot-rolled strip laminar cooling process. Journal of Process Control, 2009, 19(9): 1427—1437.
[24] Giselsson P, D Doan M, Keviczky T, Schutter B, De Rantzer A. Accelerated gradient methods and dual decomposition in distributed model predictive control. Automatica, 2013, 49(3): 829—833.
[25] Xi Y. Predictive Control. Beijing: National Defense Industry Press, 1993.
[26] Zhang Y, Li S. Networked model predictive control based on neighbourhood optimization for serially connected large-scale processes. Journal of process control, 2007, 17(1): 37—50.
[27] Dunbar WB, Murray RM. Distributed receding horizon control for multi-vehicle formation stabilization. Automatica, 2006, 42(4): 549—558.
[28] Dunbar WB. Distributed receding horizon control of dynamically coupled nonlinear systems. IEEE Transactions on Automatic Control, 2007, 52(7): 1249—1263.
[29] Farina M, Scattolini R. Distributed predictive control: a non-cooperative algorithm with neighbor-to-neighbor communication for linear systems. Automatica, 2012, 48(6): 1088—1096.
[30] Xi Y. Predictive Control 2nd Edition. Beijing: National Defense Industry Press, 2013.
[31] Zheng Y, Li S. The stabilization of coordinated distributed model predictive control. large scale complex systems theory and applications. 2013, 13(1): 182—187.
[32] Zheng Y, Li S. Distributed predictive control for building temperature regulation with impact-region optimization. Proceedings of the 19th World Congress The International Federation of Automatic Control, 2014:12074—12079.
[33] Li S, Zheng Y, Lin Z. Impacted-region optimization for distributed model predictive control systems with constraints. IEEE Transactions on Automation Science and Engineering, 2014, (99): 1—14.
[34] Zheng Y, Li S, W. Jie, Z. Xianxia. Stabilized neighborhood optimization based distributed model predictive control for distributed system. Proceedings of IEEE 31st Chinese Control Conference (CCC) 2012, 2012: 4212—4217.
[35] Keviczky T, Borrelli F, Balas GJ. Decentralized receding horizon control for large scale dynamically decoupled systems. Automatica, 2006, 42(12): 2105—2115.
[36] Motee, Nader, Sayyar-Rodsari B. Optimal partitioning in distributed model predictive control. Proceedings of IEEE American Control Conference 2003. 2003(6): 5300—5305.
[37] Mhaskar P, Gani A, Christofides PD. Fault-tolerant control of nonlinear processes: performance-based reconfiguration and robustness. International Journal of Robust and Nonlinear Control, 2006, 16(3): 91—111.
[38] Patton RJ, Clark RN, Frank PM. Issues of fault diagnosis for dynamic systems. Berlin: Springer Science & Business Media, 2000.
[39] Nash J. Non-cooperative games. Annals of mathematics, 1951: 286—295.
[40] Du X, Xi Y, Li S. Distributed model predictive control for large-scale systems. Proceedings of the IEEE 2001 American Control Conference 2001. 2001, 4: 3142—3143.
[41] Venkat AN, Rawlings JB, Wright SJ. Stability and optimality of distributed model predictive control, Procedings of the 44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. 2005: 6680—6685.
[42] Han J, Solanki SK, Solanki J. Coordinated predictive control of a wind/battery microgrid system. IEEE Journal of Emerging and Selected Topics in Power Electronics, 2013, 1(4): 296—305.
[43] Moradzadeh M, Boel R, Vandevelde L. Voltage coordination in multi-area power systems via distributed model predictive control, IEEE Transactions on Power Systems, 2013, 28(1): 513—521.
[44] Lee J-W, Hyun Kwon W, Choi J. On stability of constrained receding horizon control with finite terminal weighting matrix. Automatica, 1998, 34(12): 1607—1612.
[45] Rawlings JB, Muske KR. The stability of constrained receding horizon control. IEEE Transactions on Automatic Control, 1993, 38(10): 1512—1516.
[46] Hu T, Lin Z. Control systems with actuator saturation: analysis and design. Berilin: Springer Science & Business Media, 2001.
[47] Mayne DQ, Rawlings JB, Rao CV, Scokaert PO. Constrained model predictive control: Stability and optimality. Automatica, 2000, 36

(6): 789—814.
[48] Guan S, Wang X, Chai T. An expert controller for the laminar cooling process of hot rolled slab. Proceedings of the American Control Conference 1998. 1998, 4: 2165—2169.
[49] Zheng Y, Li S, Wang X. An approach to model building for accelerated cooling process using instance-based learning. Expert Systems with Applications, 2010, 37(7): 5364—5371.

Li Shaoyuan

Li Shaoyuan is currently, Professor, the vice-president of the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. He is also the discipline response person of Control Theory and Control Engineering, the vice director of Key Laboratory of Ministry of Education, the Vice president of Chinese Association of Automation. He received his Ph. D. degree in Computer and System Science from Nankai University in 1997. His research interests include the Predictive control, intelligent adaptive control, Fuzzy intelligent control and its applications. Five books and more than 200 papers have been published in journals/conferences, which described his researching accomplishments.



ZhengYi

ZhengYi is Associate Professor of Shanghai Jiao Tong University and currently works in the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. He received his Ph. D. degree in Control Theory and Engineering from Shanghai Jiao Tong University. He was with Shanghai Petrochemical Company, Ltd., Shanghai, from 2000 to 2003, was with GE-global Research (Shanghai) from 2010 to 2012, and was with the University of Alberta from 2014 to 2015. His research interests include the smart grid, model predictive control, system identification and their application to industrial processes.